**Project 2**

Title

**Blackjack**

Course

**CSC-17A**

Section

**46097**

Date

**July 26th, 2019**

Author

**Brandon Sanchez**

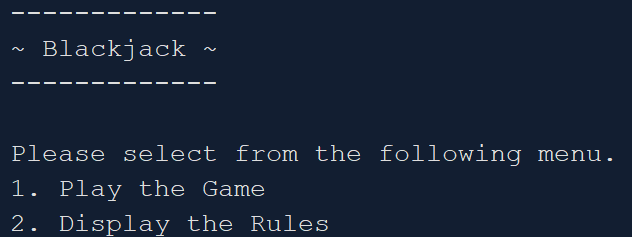
**Introduction**

Blackjack is a staple card game that is played across the world. It is simple enough for anyone to understand in one sitting yet nuanced enough for people to always come back for more. If my friends and I are playing a card game, there is a good chance that it will be Blackjack. The only table game I have played at a casino is Blackjack because that it how much I adore the game.

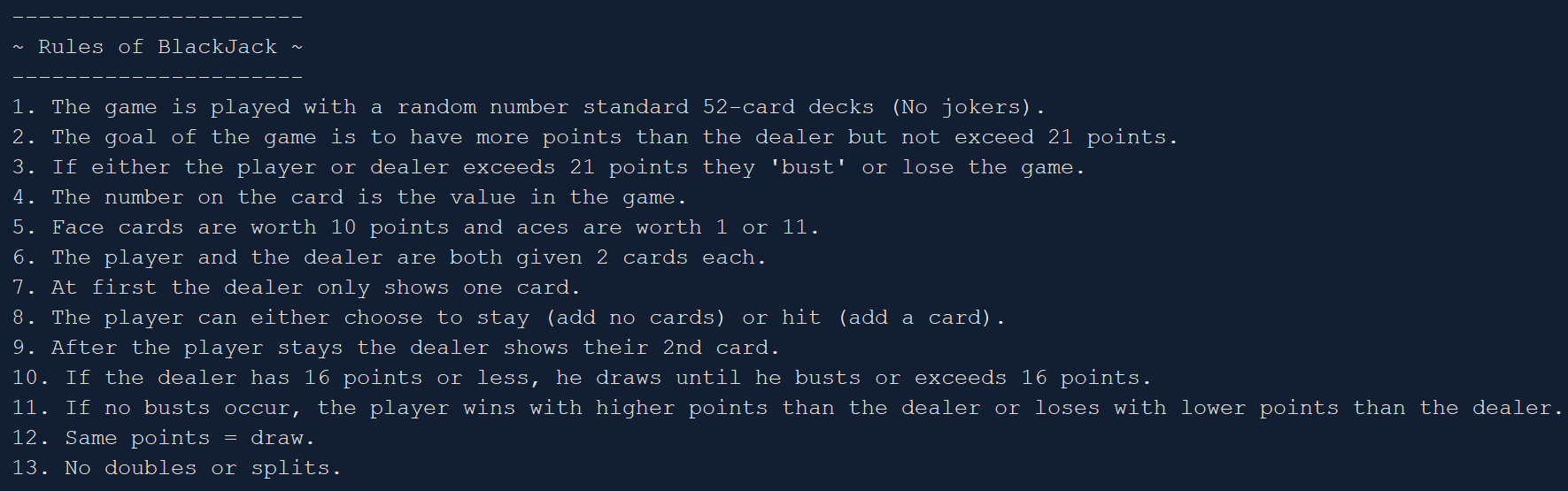
**Rules**

Available to see at the start of the game. Also, picture of rules included with sample IO.

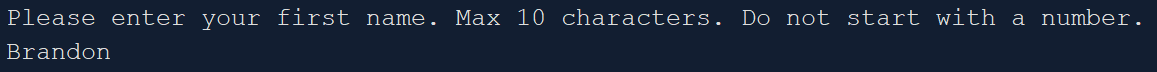
**Gameplay with Sample IO**

****

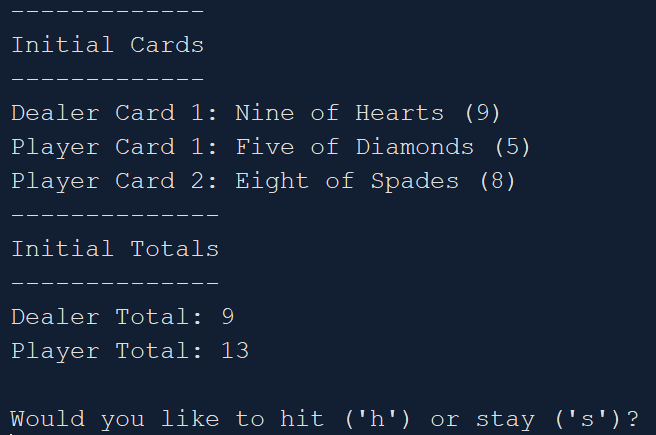
The user is greeted with this menu when they start the game.

****

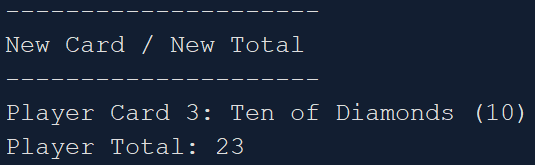
If the user inputs 2 then it will display the rules.

****

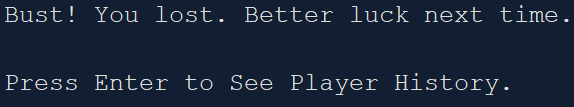
The user then inputs their name within the guidelines.

****

The initial cards/totals are outputted, and the user decides if they want to hit or stay.

****

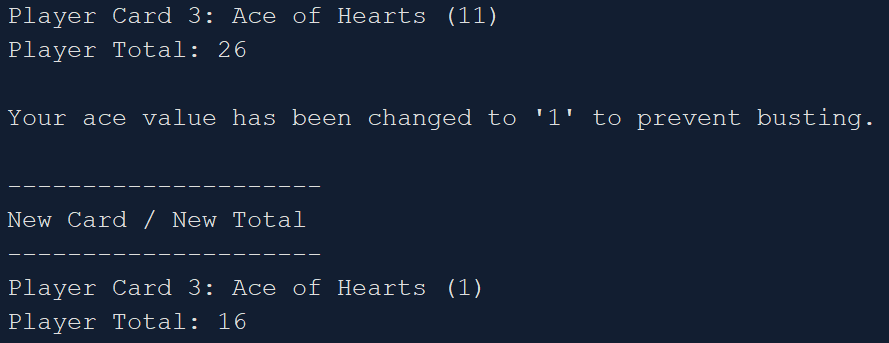
If the player hits, then it will display a new card and add it to their total.

****

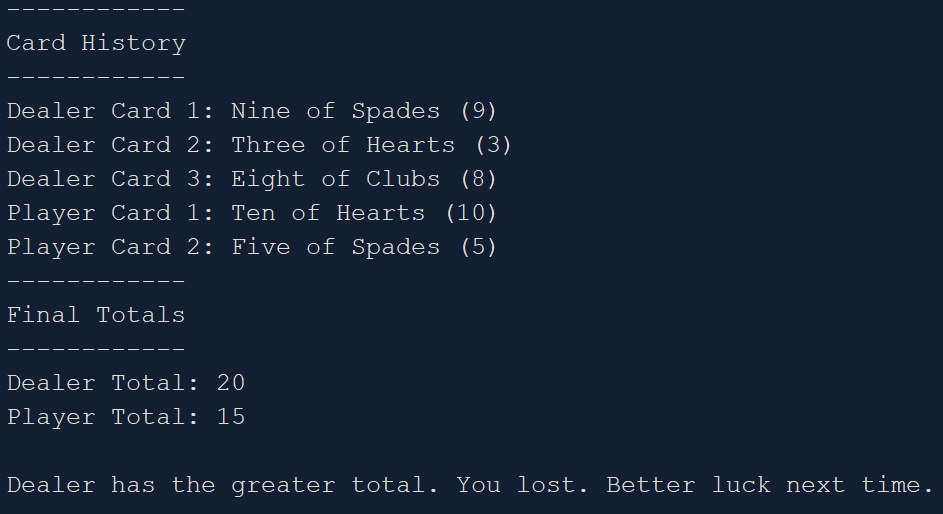
In this case the player busted because their total was greater than 21.

****

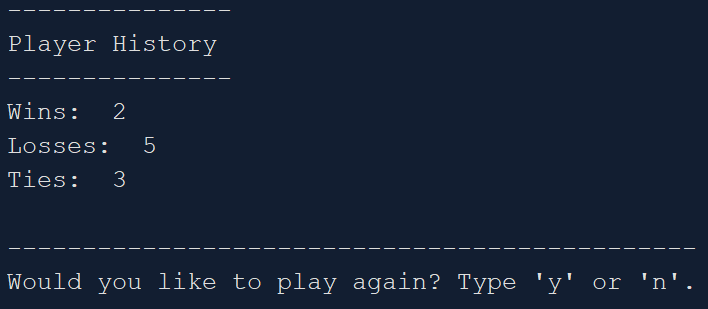
If the player is dealt an ace, then they can choose the value.

****

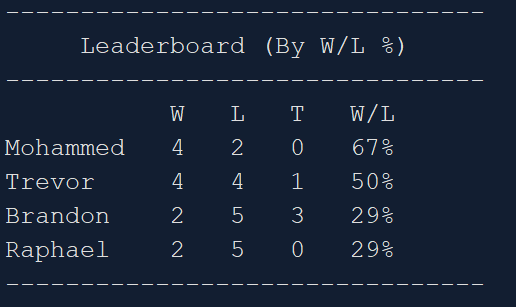
If the player has an ace worth 11 and they bust, the value will be reduced to 1 to prevent a bust.

****

If the player doesn’t bust, they will see this screen with the final results.

****

Whether busting or not, the player gets to see their match history and is prompted to play again.

****

Once the player hits ‘n’ after playing however many times they like, the program will end with a leaderboard so the player can compare their results with others. Saves their stats in a file and will be remembered for later playthroughs.

**Development Summary**

Lines of Code: 1052

I decided to carry on with the same Blackjack game project because I enjoyed working on it in CSC-5 and for project 1. The first thing I set about doing was changing all of the structures into classes. This took the majority of the time. It seems simple to change them but there are a lot of little things that change between the two. I dealt with many glitches during the process that took a lot of time to fix. I think the way my classes are setup are pretty logical and the leaderboard class could be used for many other games. The player function that holds cards and stats could also be modified to work for other card games. My favorite part of the game is the leaderboard that keeps track of each player. If your name is already in the file it will add on to your already existing wins/losses/ties. After your first playthrough it will keep track of your stats and all you have to do is play again to have the stats saved. Sadly, gameplay wise nothing has changed. There is simply not enough time for me to add all the features I want to implement because changing the base code takes so long. Hopefully if there is a game project in either CSC-17B or CSC-17C, I will continue with Blackjack and continue to make it better.

**Future Goals**

* Add doubles and splits
* Include a betting system
* Let user change ace value at any point

**Pseudocode**

Create classes.

Create function prototypes.

Set random number seed.

Declare variables and objects.

Fill leader object with player data from file.

Display menu.

If 1, play the game.

If 2, display rules then play the game.

Randomize dealer card 1.

If card = face card, card value = 10.

If card = ace, card value = 1.

Display dealer card 1.

Randomize player cards 1 & 2.

If card = face card, card value = 10.

If card = ace, card value = 1 or 11.

Display player cards 1 & 2.

If they are both worth 11, reduce 1 to a value of 1.

Display initial totals.

Ask for ‘hit’ or ‘stay’

If hit, repeat card process.

If bust, check for aces.

If there is an ace worth 11, reduce value

Else player loses, end game show history/prompt user to play again.

If stay, draw for dealer until bust or total > 16.

Display card history.

Display player/dealer totals.

Show who won.

Display player’s win history.

Prompt user to play again.

If user wants to play again loop them to start.

If user doesn’t want to play, output leaderboard and end run.